



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **04127261 A**(43) Date of publication of application: **28 . 04 . 92**

(51) Int. Cl.

**G06F 15/16**(21) Application number: **02248480**(71) Applicant: **FUJITSU LTD**(22) Date of filing: **18 . 09 . 90**(72) Inventor: **YAMAZAKI YOSHIKI****(54) MULTIPROCESSOR SYSTEM**

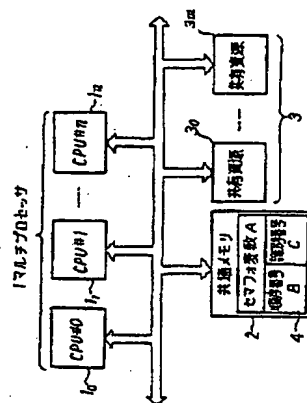
the processor having a fault can be identified.

**(57) Abstract:**

COPYRIGHT: (C)1992,JPO&amp;Japio

**PURPOSE:** To identify a processor having a fault by storing the sequence No. and the identification No. of each processor in addition to a semaphore variable, reading both sequence and identification Nos. when the shared resources are under use after reading the semaphore variable, and deciding a fault state if both Nos are identical with each other for a fixed period.

**CONSTITUTION:** A common memory 2 includes a semaphore area 4 where a sequence No. B and an identification No. C of each processor 1 are stored in addition to a semaphore variable A. Then each processor 1 reads the variable A when a request is received for use of the common resources 3 and decides whether the resources 3 are presently used or not. If not, the processor 1 writes both Nos. B and C in the area 4 and performs the due processing. Thereafter the processor 1 cancels the variable A and updates the No. B. If so, a fault state is decided for the processor of the No. C as long as both Nos B and C are identical with each other for a fixed period. Thus the prescribed fault processing is carried out. Thus a deadlock state is detected and



⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A) 平4-127261

⑬ Int.Cl.<sup>8</sup>

識別記号

庁内整理番号

⑭ 公開 平成4年(1992)4月28日

G 06 F 15/16

4 7 0 A

9190-5L

審査請求 未請求 請求項の数 3 (全6頁)

⑮ 発明の名称 マルチプロセッサシステム

⑯ 特 願 平2-248480

⑰ 出 願 平2(1990)9月18日

⑱ 発 明 者 山 崎 毅 樹 神奈川県川崎市中原区上小田中1015番地 富士通株式会社  
内

⑲ 出 願 人 富士通株式会社 神奈川県川崎市中原区上小田中1015番地

⑳ 代 理 人 弁理士 茂泉 修司

明 細 書

1. 発明の名称 マルチプロセッサシステム

2. 特許請求の範囲

(1) 共通メモリ(2)に記憶するセマフォ変数(A)を用いて共有資源(3)の排他制御を行うマルチプロセッサシステムにおいて、

該セマフォ変数に加えて各プロセッサ(1)の順序番号(B)と識別番号(C)とを記憶するセマフォ領域(4)を該共通メモリ(2)に設け、

各プロセッサ(1)が、該共有資源(3)の使用要求時に該セマフォ変数(A)を読んで該共有資源(3)が使用中でなければ使用中とし該セマフォ領域(4)に該プロセッサ(1)の順序番号(B)と識別番号(C)とを書き込んで処理を行い、処理後は該セマフォ変数(A)を解除して該順序番号(B)を更新すると共に、該セマフォ変数(A)を読んで該共有資源(3)が使用中であるときには該順序番号(B)及び該識別番号(C)を読み、両番号が一定期間中同じであれば該

識別番号(C)のプロセッサが障害状態にあると判定して所定の障害処理を行うことを特徴としたマルチプロセッサシステム。

(2) 該プロセッサ(1)の内、保守用プロセッサのみが、該共有資源(3)の使用要求時に該セマフォ変数(A)を読んで該共有資源(3)が使用中であれば該所定の障害処理を行うことを特徴とした請求項1に記載のマルチプロセッサシステム。

(3) 該プロセッサ(1)の内、保守用プロセッサ以外のプロセッサにおいては、該共有資源(3)の使用要求時に該セマフォ変数(A)を読んで該共有資源(3)が使用中であれば該所定の障害処理を行わずに該共有資源(3)が使用中でなくなるまで待機することを特徴とした請求項1又は2に記載のマルチプロセッサシステム。

3. 発明の詳細な説明

(概 要)

共有資源の排他制御を行うマルチプロセッサシステムに関し、

共通メモリに記憶するセマフォ変数を用いて共有資源の排他制御を行うマルチプロセッサシステムにおいて、デッド・ロック状態を検出して障害発生のプロセッサを識別できるようにすることを目的とし、

該セマフォ変数に加えて各プロセッサの順序番号と識別番号とを記憶するセマフォ領域を該共通メモリに設け、各プロセッサが、該共有資源の使用要求時に該セマフォ変数を読んて該共有資源が使用中でなければ使用中とし該セマフォ領域に該プロセッサの順序番号と識別番号とを書き込んで処理を行い、処理後は該セマフォ変数を解除して該順序番号を更新すると共に、該セマフォ変数を読んて該共有資源が使用中であるときには該順序番号及び該識別番号を読み、両番号が一定期間中同じであれば該識別番号のプロセッサが障害状態にあると判定して所定の障害処理を行うように構成する。

しており、20はこれらマルチプロセッサの共通メモリ、そして3。〜3。は $m+1$ 個の各種共有資源（ディスク装置、プリンタ、I/O装置等）であり、以下総称して3で示す。

このようなマルチプロセッサシステムにおいては、共有資源3を排他的に使用させるため、共通メモリ20にセマフォ変数の領域21を設け、次のような制御を行っている。

- ①共通メモリ20のセマフォ変数Aが“0”のときは共有資源3が未使用、“1”のときが使用中を示しているとする、共有資源3の使用要求を行うプロセッサは、セマフォ変数Aを読んて共有資源3が使用中か否かを判定する。
- ②その結果、未使用（A=0）であるなら、A=1にセットして他のプロセッサの使用を禁止し、当該共有資源3の使用権を確保する。
- ③当該共有資源3の使用が終了したときには、A=0として使用中を解除し、他のプロセッサに対して使用権を開放する。

#### 【産業上の利用分野】

本発明は、マルチプロセッサシステムに関し、特に共有資源の排他制御を行うマルチプロセッサシステムに関するものである。

近年、情報処理量の増大に伴って1つのプロセッサ（CPU）で全体を処理する代わりに、複数のプロセッサで分担して処理するマルチプロセッサシステムが用いられているが、この場合、複数のプロセッサが共有資源としての例えばプリンタに同時印字を行うと、用紙上で文字が混在してしまう。

これを防止するため、実際には1つのプロセッサが共有資源を排他的に確保できるようにする必要がある。

#### 【従来の技術】

第6図は、従来のマルチプロセッサシステムを概略的に示したもので、図中、10。〜10。は共通バスBに接続された $n+1$ 個のプロセッサ（CPU#0〜# $n$ ）でマルチプロセッサを構成

#### 【発明が解決しようとする課題】

このような従来例の場合には、共有資源の使用権を確保したプロセッサがソフトウェア・バグ等の障害により誤動作した場合には、その共有資源3は他のプロセッサが永久に使用できなくなってしまうデッド・ロック状態を呈し、1つのプロセッサの障害がマルチプロセッサシステム全体の障害を引き起こす危険性がある。

しかしながら、このような障害状態が発生しても、他のプロセッサにとっては障害状態か否か、またどのプロセッサに障害が発生しているのか不明であり、障害に対処できないという問題点があった。

従って、本発明は、共通メモリに記憶するセマフォ変数を用いて共有資源の排他制御を行うマルチプロセッサシステムにおいて、上記のようなデッド・ロック状態を検出して障害発生のプロセッサを識別できるようにすることを目的とする。

## 〔課題を解決するための手段及び作用〕

第1図は、上記の課題を解決するための本発明に係るマルチプロセッサシステムの構成を原理的に示したもので、本発明では、セマフォ変数Aに加えて各プロセッサ1。～1。を総称して示すマルチプロセッサ1の各々の順序番号Bと識別番号Cとを記憶するセマフォ領域4を共通メモリ2に設けている。

そして、各プロセッサ1は、第2図のフローチャートに示すように、共有資源3の使用要求時にセマフォ変数Aを読み（第2図のステップS1）、共有資源3が使用中か否かを判定し（同S2）、使用中でなければ使用中とし、セマフォ領域4に該プロセッサ1の順序番号Bと識別番号Cとを書き込んで（同S3）処理を行い（同S4）、処理後はセマフォ変数Aを解除（同S5）して順序番号Bを更新する（同S6）。

一方、セマフォ変数Aを読んで該共有資源3が使用中であるときには順序番号B及び識別番号Cを読み、両番号が一定期間中同じであれば（同S

7）、識別番号Cのプロセッサが障害状態にあると判定して所定の障害処理を行う（同S8）ようにしている。

このようにして、本発明では、共有資源3を使用しているプロセッサが長い間同じものであるときには、そのプロセッサに何らかの障害が発生しているものと判定することができ、所定の障害処理を行うことができることになる。

また、本発明では、プロセッサ1の内、保守用プロセッサのみが、共有資源3の使用要求時に該セマフォ変数Aを読んで共有資源3が使用中であれば所定の障害処理を行うようにしてもよい。

更に、本発明では、プロセッサ1の内、保守用プロセッサ以外のプロセッサにおいては、共有資源3の使用要求時にセマフォ変数Aを読んで共有資源3が使用中であれば所定の障害処理を行わずに共有資源3が使用中でなくなるまで待機するようにしてもよい。

## 〔実施例〕

第3図は、第1図に示した本発明に係るマルチプロセッサシステムにおける各プロセッサのプログラム動作の実施例を示したフローチャート図であり、第2図と同じ符号のステップS1～S6は同じ処理工程を示している。

そして、この実施例では、ステップS1でセマフォ変数Aを共通メモリ2のセマフォ領域4から読み出す前に、ステップS0において共有資源3の使用要求が或るプロセッサ1から発生されたとき、そのプロセッサ1において制御変数CNTを1に初期化する。これは、例えばカウンタ（リトライカウンタ）を“1”にすることによって行われる。

そして、セマフォ変数Aが“0”で別のプロセッサが共有資源3を使用していなかったことがステップS2で判明したときには、第2図の場合と同様にしてステップS3～S6が実行されるが、セマフォ変数Aが“1”で別のプロセッサが共有資源3を使用していることがステップS2で判明

したときには、第2図のステップS7に対応する以下のステップS71～S75が実行される。

即ち、まず、順序番号Bと識別番号Cとをセマフォ領域4からプロセッサが読み込み、前面に読み込んだB、Cの値と比較する（同S71）。

この結果、一致したときには、前面のアクセス時と同じプロセッサが共有資源3を使用しているので、上記の制御変数CNTを例えば“1”だけインクリメントする（同S72）。そして、このインクリメントした結果の制御変数CNTの値が一定時間（この場合は一定回数）を示すリトライアウト時間Rと比較する（同S73）。

その結果、CNT≦Rと判明したときには、タイマセットを行い（同S75）、ステップS1に戻る。

また、順序番号Bと識別番号Cとをセマフォ領域4からプロセッサが読み込み、前面に読み込んだB、Cの値と比較して（同S71）一致しなかったときには、前面のアクセス時とは別のプロセッサ、若しくは前面アクセス時と同一のプロセッサ

サ一が別の処理において共有資源3を使用している  
るので、単に自局のアクセス・タイミングが悪か  
ったとして制御変数CNTを"1"に初期化して  
タイマセットしステップS1に進む。

このようなルーチンを何回か繰り返したときに  
ステップS71とS72を常に経過してステップ  
S73に到り、しかもリトライアウト時間R<C  
NTとなったときには、或るプロセッサがこのリ  
トライアウト時間Rの間ずっと共有資源3を占有  
していたことになるので、そのプロセッサは障害  
が発生していると判断し、ステップS9で障害処  
理を実行する。尚、全ての処理において、共有資  
源をアクセスする際、その占有時間を一定以下に  
する必要がある。また、この障害処理例としては、  
該当共有資源を強制的に開放し、該当プロセッサ  
を閉塞させる等が挙げられる。

第4図は、第3図に示した各プロセッサに共通  
の処理の内、障害状態を発見して対処させるのを  
容易にするため、或る特定のプロセッサを保守用  
に予め決めておき、そのプロセッサのみが実行す

号のプロセッサが障害状態にあると判定して所定  
の障害処理を行うように構成したので、個々のプ  
ロセッサの誤動作に基づくデッドロック状態を発  
見して対処することができ、システム全体の信頼  
性が大きく向上することとなる。

#### 4. 図面の簡単な説明

第1図は、本発明に係るマルチプロセッサシ  
ステムの原理構成ブロック図、

第2図は、本発明に係るマルチプロセッサシ  
ステムの動作を原理的に説明するためのフローチャ  
ート図、

第3図は、本発明に係るマルチプロセッサシ  
ステムにおける各プロセッサの動作実施例を示した  
フローチャート図、

第4図は、本発明に係るマルチプロセッサシ  
ステムにおける保守用プロセッサの動作実施例を示  
したフローチャート図、

第5図は、本発明に係るマルチプロセッサシ  
ステムにおける非保守用プロセッサの動作実施例を

るプログラムを示したもので、それぞれのステッ  
プは第3図に示したものと同じであるが、第3図  
と異なるのは、共有資源非使用中の判定(ステッ  
プS2)後は制御変数CNTを初期化する(同S  
74)ことである。

第5図は、第3図に示した各プロセッサに共通  
の処理の内、第4図の保守用プロセッサ以外のプ  
ロセッサが実行するプログラムを示したもので、  
それぞれのステップは第3図に示したものと同じ  
であるが、第3図と異なるのは、共有資源使用中  
の判定(ステップS2)後は一定時間待機(同S  
10)してからステップS1に戻る点である。

#### (発明の効果)

以上のように本発明に係るマルチプロセッサシ  
ステムによれば、セマフォ変数に加えて各プロセ  
ッサの順序番号と識別番号とを別途共通メモリに  
記憶させ、各プロセッサが、共有資源の使用要求  
時に共有資源が同じプロセッサの同じ処理によっ  
て或る期間中占有されている場合にはその識別番

示したフローチャート図、

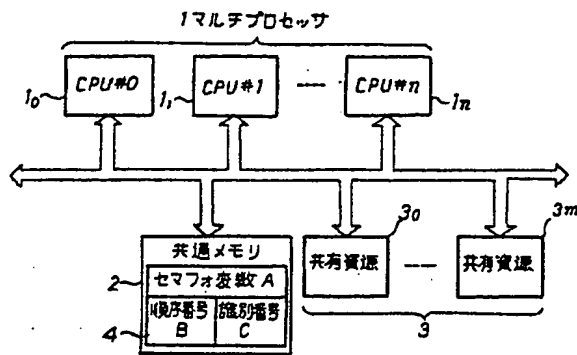
第6図は、従来例を示す図、である。

第1図において、

- 1(1.~1.)…プロセッサ、
- 2…共通メモリ、
- 3(3.~3.)…共有資源、
- 4…セマフォ領域、
- A…セマフォ変数、
- B…順序番号、
- C…識別番号。

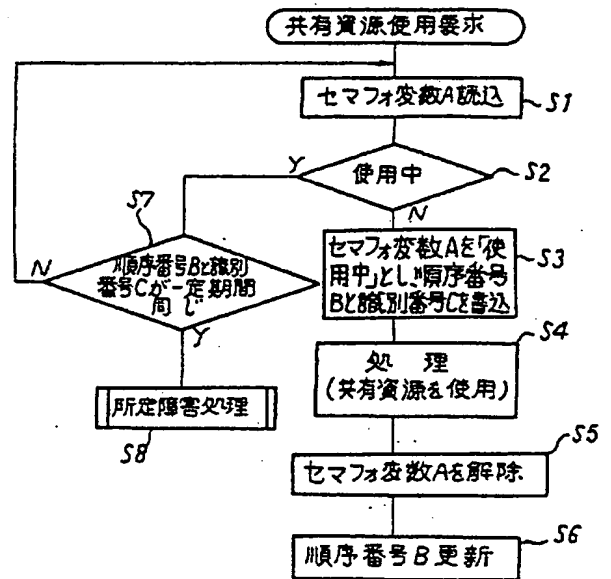
図中、同一符号は同一又は相当部分を示す。

代理人 弁理士 茂 泉 修 司



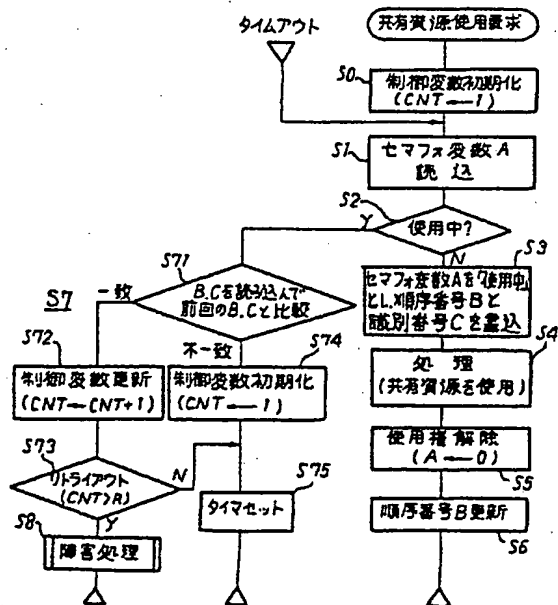
本発明の原理構成図

第 1 図



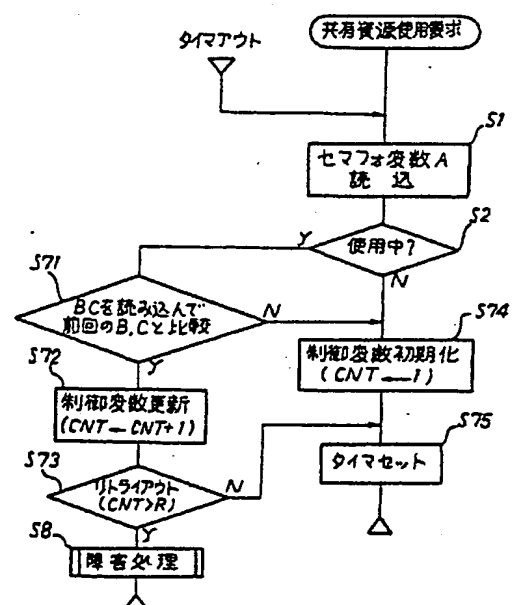
本発明の原理作用図

第 2 図



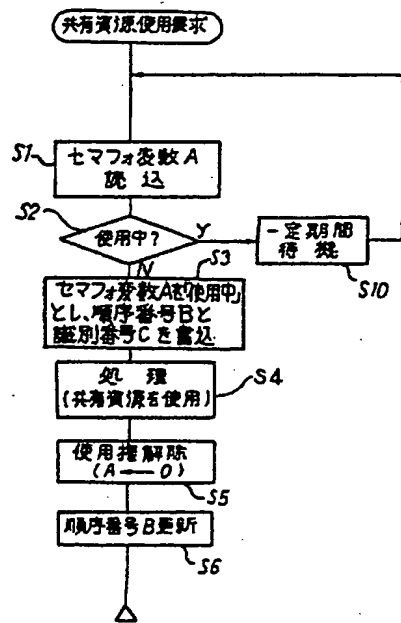
本発明の動作例

第 3 図



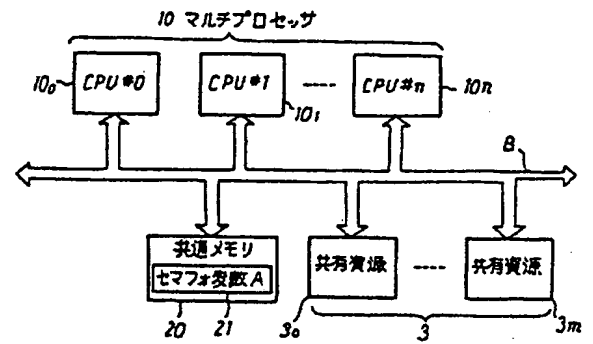
保守用プロセッサの動作例

第 4 図



低守用以外のプロセッサの動作例

第 5 図



従 来 例

第 6 図